11. Изрази. Приоритет и асоциативност на операторите (обобщение)

Проф. д-р Емил Хаджиколев

1. Други оператори (оператори за присвояване и др.)
2. Изрази;
3. Приоритет на операторите;
4. Преобразуване на типове;
5. Асоциативност.

Оператори за присвояване в C++

- Освен стандартния оператор за присвояване, има и други оператори, свързани с двуаргументните аритметични и побитови оператори.

  `+= -= *= /= %= &= |= ^= <<= >>=`

- Левият операнд е променлива, в която се записва резултата от изпълнението на аритметичния/побитов оператор заедно с десния операнд.
- Действието на сложните оператори за присвояване може да се изрази чрез стандартни оператори. Напр.,

  `i += 3;` е равносилно на `i = i + 3;`

  `i %= 3;` е равносилно на `i = i%3;`

Други оператори в C++

- За пълнота на изложението, в следващата таблица са описани и други основни оператори, някои от които ще разглеждаме в други лекции.

Оператор	Име	Описание
a[b]	Достъп до елемент на масив	Пример: arr[0], arr[1]. Преди скобите се записва име на масив, а в скобите се задава индекс на елемент на масива. Резултатът е от типа на елементите на масива.
*a, a.b, a->b, a.*b, a->*b		Оператори за достъп до статични и динамични елементи
&a	адрес на	Оператор, връщащ като резултат адрес на обект или функция.
(тип)	Преобразуване по тип	Преобразува операнд до указан тип. Може да се използва само за съвместими типове.
(параметри)	Обръщение към функция	В скобите се задават фактически параметри, а преди тях - име на метод. Резултатът може да е от всякакъв тип.
new, new[]	Създаване на динамичен обект	Резултатът е референция към създаден динамичен обект.
delete, delete[]	Унищожаване на динамичен обект	Унищожава един динамичен обект или динамичен масив.

Изрази

- **Комбинацията от няколко оператора и операнда се нарича израз**.
- Има прости изрази, състоящи се от един оператор или дори само операнд (литерал, променлива, константа или функция, която връща стойност) и сложни изрази, съдържащи повече от един оператор.
- Възможно е в един израз да участват операнди от различни типове. Как точно се изчислява един сложен израз и какъв е крайният му тип, се определя от приоритета и асоциативността на участващите оператори.

Приоритет на операторите

- В израз операторите, които се изпълняват първо са с най-висок приоритет, след това операторите с по-нисък и т.н…, а накрая – с най-нисък.
- **Приоритетът определя реда на изпълнение (прилагане) на операторите в израз**.
- Използването на скоби, указва изразите в тях да се изчисляват с предимство.

Асоциативност на операторите

- **Асоциативността определя реда на изчисление на операндите за оператор.**
- Повечето оператори са ляво-асоциативни. При тях първо се изчислява левият операнд, а след това – десният, и накрая се извършва действието предвидено от оператора.
- При дясно-асоциативните, първо се изчислява десният операнд, след него левият, и накрая се прилага оператора.

Приоритет и асоциативност на операторите в C++ (https://cppreference.com/w/cpp/language/operator_precedence.html) (1)

Приоритет	Оператор	Описание	Асоциативност
1	a::b	Scope resolution	Left-to-right →
2	a++ a--	Suffix/postfix increment and decrement	
	type(a) *type*{a}	Functional cast	
	a()	Function call	
	a[]	Subscript	
	a.b a->b	Member access	
3	++a --a	Prefix increment and decrement	Right-to-left ←
	+a -a	Unary plus and minus	
	!a ~a	Logical NOT and bitwise NOT	
	(*type*)a	C-style cast	
	*a	Indirection (dereference)	
	&a	Address-of	
	sizeof	Size-of[note 1]	
	co_await	await-expression (C++20)	
	new – new[]	Dynamic memory allocation	

Приоритет и асоциативност на операторите в C++ (https://cppreference.com/w/cpp/language/operator_precedence.html) (2)

Приоритет	Оператор	Описание	Асоциативност
4	a.*b a->*b	Pointer-to-member	Left-to-right →
5	a * b a / b a % b	Multiplication, division, and remainder	
6	a + b a - b	Addition and subtraction	
7	a << b a >> b	Bitwise left shift and right shift	
8	a <=> b	Three-way comparison operator (since C++20)	
9	a < b a <= b a > b a >= b	For relational operators **<** and **<=** and **>** and **>=** respectively	
10	a == b a != b	For equality operators **=** and **!=** respectively	
11	a & b	Bitwise AND	
12	a ^ b	Bitwise XOR (exclusive or)	
13	a | b	Bitwise OR (inclusive or)	
14	a && b	Logical AND	
15	a || b	Logical OR	

Приоритет и асоциативност на операторите в C++ (https://cppreference.com/w/cpp/language/operator_precedence.html) (3)

Приоритет	Оператор	Описание	Асоциативност
16	a ? b : c	Ternary conditional[note 2]	Right-to-left ←
	throw	throw operator	
	co_yield	yield-expression (C++20)	
	a = b	Direct assignment (provided by default for C++ classes)	
	a += b a -= b	Compound assignment by sum and difference	
	a *= b a /= b a %= b	Compound assignment by product, quotient, and remainder	
	a <<= b a >>= b	Compound assignment by bitwise left shift and right shift	
	a &= b a ^= b a |= b	Compound assignment by bitwise AND, XOR, and OR	
17	a, b	Comma	Left-to-right →